In the Claims:

Claims 1-51 were pending at the time of the Office Action.

Claims 1-4, 7-10, 12, 13, 16-21, 23, 24, 26-28, 31-34, 37, 38, 40-42, 45-48 and 51 are rejected.

Claims 5, 6, 11, 14, 15, 22, 25, 29, 30, 34, 35-36, 39, 43, 44, 49, and 50 are objected to.

No claims are canceled or amended by the current response.

Accordingly, claims 1-51 remain pending and are provided herein below in a complete listing of claims for the sake of convenience.

Listing of Claims:

1. (Original) An Internet protocol (IP) filter, comprising processor-executable instructions that, when executed on a processor, perform the following steps:

monitoring Internet protocol data packets transmitted from one or more clients to a server;

obtaining a network address from an IP data packet transmitted by a client; and

processing IP data packets from the client if a Network address that is uniquely associated with the client is stored in a client table.

2. (Original) The Internet protocol filter as recited in claim 1, further comprising processor-executable instructions that, when executed on a processor, perform the following steps:

if the Network address is not stored in the client table, retrieving a client limit value from a client limit field, the client limit value indicating a maximum number of unique clients for which IP data packets can be processed;

processing IP data packets from the client if the number of Network addresses in the client table is less than the client limit value; and

storing the Network address in the client table.

3. (Original) The Internet protocol filter as recited in claim 1, wherein the client is a first client and the Network address is a first Network address, the Internet protocol filter further comprising processor-executable instructions that, when executed on a processor, perform the following steps:

if the first Network address is not stored in the client table, retrieving a client limit value from a client limit field, the client limit value indicating a maximum number of unique clients for which IP data packets can be processed;

if the number of Network addresses in the client table is greater than or equal to the client limit value, determining if the first client is represented in the client table by a second Network address that is different from the first Network address; and

processing IP data packets from the first client if the second Network address is found in the client table.

4. (Original) The Internet protocol filter as recited in claim 3, further comprising processor-executable instructions that, when executed on a processor, perform the following steps:

removing the second Network address from the client table; and inserting the first Network address into the client table.

5. (Original) The Internet protocol filter as recited in claim 1, further comprising processor-executable instructions that, when executed on a processor, perform the following steps:

if the first Network address is not stored in the client table, retrieving a client limit value from a client limit field, the client limit value indicating a maximum number of unique clients for which IP data packets can be processed;

if the number of Network addresses in the client table is greater than or equal to the client limit value, transmitting a signal to each Network address listed in the client table; and

if a client at a second Network addresses does not respond to the signal, removing the second Network address from the client table, inserting the first Network address into the client table and processing IP data packets from the first client.

6. (Original) The Internet protocol filter as recited in claim 5, further comprising processor-executable instructions that, when executed on a processor, perform the following steps:

removing the second Network address from the client table; and

inserting the first Network address into the client table.

7. (Original) A method, comprising:

detecting when a current client attempts to establish a connection with a server;

determining a unique client identifier that is associated with the current client;

determining if a total number of previous clients having access to the server is less than a client limit;

determining if the current client has previously been allowed to access the server;

providing access to the server if the total number of previous clients having access to the server is less than a client limit;

providing access to the server if the total number of previous clients is greater than or equal to the client limit and if the current client has previously been allowed to access the server; and

storing the unique client identifier associated with the current client in memory if access is provided to the current client.

8. (Original) The method as recited in claim 7, wherein the determining if a total number of previous clients having access to the server is less than a client limit further comprises:

determining how many unique identifiers are stored in memory; and

comparing the number of unique identifiers in memory with the client limit.

9. (Original) The method as recited in claim 7, wherein the determining if the current client has previously been allowed to access the server is only performed if the total number of previous clients having access to the server is greater than or equal to the client limit.

10. (Original) The method as recited in claim 7, wherein the determining if the current client has previously been allowed to access the server further comprises:

comparing the unique identifier of the current client with the unique identifiers of each previous client that has been allowed to access the server;

determining that the current client has previously been allowed to access the server if the current client identifier matches a previous client identifier.

11. (Original) The method as recited in claim 7, wherein the determining if the current client has previously been allowed to access the server further comprises:

transmitting a signal to each previous client that has been allowed to access the server; and

determining that the current client has previously been allowed to access the server if at least one of the previous clients fails to acknowledge the signal.

12. (Original) The method as recited in claim 7, further comprising:

pre-configuring the client limit; and

storing the client limit in memory.

13. (Original) The method as recited in claim 12, wherein the client limit has a pre-defined maximum to which it may be configured.

14. (Original) The method as recited in claim 7, further comprising:

pre-configuring the client limit;

encrypting the client limit; and

storing the encrypted client limit in memory.

15. (Original) The method as recited in claim 7, further comprising:

retrieving an encrypted client limit; and

decrypting the encrypted client limit to derive the client limit.

16. (Original) The method as recited in claim 7, wherein the determining the unique client identifier that is associated with the current client further comprises identifying an Internet protocol address from a data packet transmitted by the current client.

17. (Original) The method as recited in claim 7, further comprising storing the unique client identifiers in a client table in memory.

18. (Original) The method as recited in claim 7, wherein the client identifier is a network address.

19. (Original) A server that provides access to a limited number of clients, comprising:

memory;

a network interface configured to handle communications between the server and a plurality of clients;

an operating system stored in the memory;

a client limit stored in the memory, the client limit denoting a number of unique clients that are allowed to access the server;

an IP stack in the memory that is used to process data packets transmitted from clients;

a client table in the memory for storing a unique Network address for each client that accesses the server; and

a communications filter configured to allow access to a first client if the total number of clients that have accessed the server is less than the client limit, or if the total number of clients that have accessed the server is greater than or equal to the client limit and the first client has previously accessed the server.

20. (Original) The server as recited in claim 19, wherein the Communications filter is further configured to search the client table for a first Network address associated with the first client and determine that the first client has previously accessed the server if the first Network address is found in the client table.

21. (Original) The server as recited in claim 19, wherein the Communications filter is further configured to search the client table for a second Network address associated with the first client and determine that the first client has previously accessed the server if the second Network address is found in the client table.

22. (Original) The server as recited in claim 21, wherein the Communications filter is further configured to determine the second Network

address by signaling each Network address listed in the client table and determine that the second Network address is a network address listed in the table that does not acknowledge the signal.

23. (Original) The server as recited in claim 19, wherein the client limit is configurable.

24. (Original) The server as recited in claim 19, wherein the Communications filter is further configured to signal that the client limit has been exceeded and to deny server access to the first client if the total number of clients that have accessed the server is greater than or equal to the client limit, and the first client has not previously accessed the server.

25. (Original) The server as recited in claim 19, wherein the client limit is encrypted, the server further comprising a decryption module configured to decrypt the encrypted client limit.

26. (Original) A method for providing server access to a limited number of clients, the method comprising:

monitoring TCP/IP packets sent from a plurality of clients to a server;

obtaining a unique Network address for each client from one or more packets transmitted by the client;

storing the Network address of each client that accesses the server;

determining if a client limit has been reached; and providing access to a first client if the client limit has been reached, or if the first client has previously accessed the server.

27. (Original) The method as recited in claim 26, wherein the determining if the client limit has been reached further comprises:

determining how many unique clients have accessed the server;

comparing the number of unique client with the client limit; and

determining that the client limit has been reached if the number of unique clients is greater than or equal to the client limit.

28. (Original) The method as recited in claim 26, wherein the determining if the first client has previously accessed the server further comprises:

comparing a first Network address that uniquely identifies the first client with a table of stored Network addresses; and

determining that the first client has previously accessed the server if the first Network address matches a stored Network address.

29. (Original) The method as recited in claim 26, wherein the determining if the first client has previously accessed the server further comprises:

sending a signal to each of multiple Network addresses of clients that have accessed the server; and

if there is no response to one of the signals, determining that the first client has previously accessed the server using the Network address of the client from which there was no response detected.

30. (Original) The method as recited in claim 26, further comprising:

retrieving an encrypted client limit; and

decrypting the encrypted client limit to derive the client limit.

31. (Original) An operating system stored on a computer-readable medium, the operating system comprising:

an IP stack for processing Internet protocol data packets received from multiple clients;

a client limit field containing a client limit value that denotes a maximum number of clients that may access the IP stack;

a client table containing a unique Network address for each client that has accessed the operating system; and

a communications filter configured to determine a first Network address of a first client attempting to access the operating system, search the client table for the first Network address, and allow the first client to access the operating system if the first Network address is found in the client table.

32. (Original) The operating system as recited in claim 31, wherein the Communications filter is further configured to:

allow the first client to access the operating system if the number of Network addresses in the client table is less than the client limit value; and

store the first Network address in the client table if the first client is allowed to access the operating system.

33. (Original) The operating system as recited in claim 31, wherein the Communications filter is further configured to allow the first client to access the operating system if the number of Network addresses in the client table is greater than or equal to the client limit value and the first client has previously accessed the operating system using a second Network address that is stored in the client table.

34. (Original) The operating system as recited in claim 33, wherein the Communications filter is further configured to:

transmit a signal to each Network address listed in the client table;

monitor for an acknowledgement to each signal; and

if an acknowledgement is not received from a network address in the client table, determining that the non-acknowledging Network address is the second Network address used by the first client.

35. (Original) The operating system as recited in claim 34, wherein the Communications filter is further configured to replace the second Network address in the client table with the first Network address.

36. (Original) The operating system as recited in claim 31, wherein the client limit value is encrypted, and the operating system further comprises a decryption module that is configured to decrypt the client limit value.

37. (Original) A computer-readable medium comprising computer-executable instructions that, when executed on a computer, perform the following steps:

determining a first Internet Protocol (IP) address transmitted from a first client to a server;

searching a client table for the first Network address; and

allowing the first client to access the server if the first Network address is found in the client table.

38. (Original) The computer-readable medium as recited in claim 37, further comprising computer-executable instructions that, when executed on a computer, perform the following steps:

determining if a client limit has been reached, the client limit indicating a total number of clients that can access the server;

allowing the first client to access the server if the client limit has not been reached; and

inserting the first Network address into the client table.

39. (Original) The computer-readable medium as recited in claim 37, further comprising computer-executable instructions that, when executed on a computer, perform the following steps:

transmitting a signal to each Network address listed in the client table; and

if there is no response from one of the Network addresses signaled, allowing the first client to access the server, removing the non-responsive Network address from the client table, and inserting the first Network address into the client table.

40. (Original) A computer system, comprising:

a processor;

a network interface card to handle communications with multiple clients;

memory;

a global system registry;

a client table having one entry for each client allowed to access the system, each entry including a unique Internet protocol (IP) address for each client; and

a communications filter configured to:

retrieve a client limit from the global system registry;

determine a first Network address that is associated with a first client attempting to access the system;

allow the first client to access the system if the first Network address is stored in the client table or if the number of client table entries is less than the client limit; and

store the first Network address in the client table if the first client is allowed to access the system.

41. (Original) The computer system as recited in claim 40, wherein the Communications filter is further configured to allow the first client to access the system if the number of entries in the client table is greater than or equal to the client limit and if the first client has previously accessed the system.

42. (Original) The computer system as recited in claim 41, wherein the Communications filter is further configured to determine if the first client has previously accessed the system if the first Network address is stored in the client table.

43. (Original) The computer system as recited in claim 41, wherein the Communications filter is further configured to determine if the first client has previously accessed the system by transmitting a signal to each Network address listed in the client table, monitoring responses to the signals to determine if a client at a second Network address is no longer using the Network address, substituting the first Network address in the table for the second Network address and allowing the first client to access the system if the client at the second Network address does not respond to the signal.

44. (Original) The computer system as recited in claim 40, wherein the client limit is encrypted, the computer system further comprising a decryption module configured to decrypt the encrypted client limit.

45. (Original) A communications protocol filter, comprising processor-executable instructions that, when executed on a processor, perform the following steps:

monitoring communications protocol data packets transmitted from one or more clients to a server;

obtaining a network address from a communications protocol data packet transmitted by a client; and processing communications protocol data packets from the client if a Network address that is uniquely associated with the client is stored in a client table.

46. (Original) The communications protocol filter as recited in claim 45, further comprising processor-executable instructions that, when executed on a processor, perform the following steps:

if the Network address is not stored in the client table, retrieving a client limit value from a client limit field, the client limit value indicating a maximum number of unique clients for which communications data packets can be processed;

processing communications protocol data packets from the client if the number of Network addresses in the client table is less than the client limit value; and

storing the Network address in the client table.

47. (Original) The communications protocol filter as recited in claim 45, wherein the client is a first client and the Network address is a first Network address, the communications protocol filter further comprising processor-executable instructions that, when executed on a processor, perform the following steps:

if the first Network address is not stored in the client table, retrieving a client limit value from a client limit field, the client limit value indicating a maximum number of unique clients for which communications protocol data packets can be processed;

if the number of Network addresses in the client table is greater than or equal to the client limit value, determining if the first client is represented in the client table by a second Network address that is different from the first Network address; and

processing communications protocol data packets from the first client if the second Network address is found in the client table.

48. (Original) The communications protocol filter as recited in claim 47, further comprising processor-executable instructions that, when executed on a processor, perform the following steps:

removing the second Network address from the client table; and

inserting the first Network address into the client table.

49. (Original) The communications protocol filter as recited in claim 45, further comprising processor-executable instructions that, when executed on a processor, perform the following steps:

if the first Network address is not stored in the client table, retrieving a client limit value from a client limit field, the client limit value indicating a maximum number of unique clients for which communications protocol data packets can be processed;

if the number of Network addresses in the client table is greater than or equal to the client limit value, transmitting a signal to each Network address listed in the client table; and

if a client at a second Network addresses does not respond to the signal, removing the second Network address from the client table, inserting the first

Network address into the client table and processing communications protocol data packets from the first client.

50. (Original) The communications protocol filter as recited in claim 49, further comprising processor-executable instructions that, when executed on a processor, perform the following steps:

removing the second Network address from the client table; and

inserting the first Network address into the client table.

51. (Original) The communications protocol filter as recited in claim 45, wherein the communications protocol is an Internet protocol and the communications protocol data packets are Internet protocol data packets.